**API DOCUMENTATION**

This API serves TF-Models developed in the ZIEL-Project.

1. The first model detects and classifies an image (object detection). The development and the validation is peer-review published:

<https://www.researchgate.net/publication/357893411_Development_and_Evaluation_of_a_Bayesian_Risk_Stratification_Method_for_Major_Amputations_in_Patients_with_Diabetic_Foot_Ulcers?_sg%5B1%5D=>

1. The second model classifes a cropped image according to its maceration status. It is published here: <https://www.researchgate.net/publication/357891635_Automatic_Classification_of_Diabetic_Foot_Ulcer_Images_-_A_Transfer-Learning_Approach_to_Detect_Wound_Maceration?_sg%5B1%5D=>

The models development and validation process was published on two international conferences.

Folder Tree (tf-serving):

* tf-serve/ → Tensorflow serving for Classification Model
* web/ → Flask API
* docker-compose.yml

Requirements:

* docker
* docker-compose

Running program using:

$ sudo docker-compose up --build -d

There is a demo script to test the API function:

1. $
2. $ python test-gradientcam.py

API:

Get Maceration Status

|  |  |
| --- | --- |
| BASE\_URL | localhost:8080 |
| ENDPOINT | https://BASE\_URL/predict |
| Description | Run prediction for given image to look for maceration |
| Method | POST |
| Body | image → image file |
| Path Parameter | python test-classification.py |
| Example Request | curl -X POST -F image=@images/maceration-present-48393.png http://127.0.0.1:5000/image/predict |
| Example Response | **Success**:  {  "inference": {  "label": "Maceration Absent",  "probability": 0.0586852431  }  } |
| Response Notes | label → predicted label (Maceration Absent if probability < 0.5 and Maceration Present if probability >= 0.5  probability -> Model’s inference on the probability that a maceration is present in the image |

Get Gradientcam

|  |  |
| --- | --- |
| BASE\_URL | localhost:8080 |
| ENDPOINT | https://BASE\_URL/predict |
| Description | Run prediction and add heatmap (gradientcam) to visualize salient regions in the image that trigger classification |
| Method | POST |
| Body | image → image file |
| Path Parameter | python test-classification.py |
| Example Request | curl -X POST -F image=@images/maceration-present-48393.png http://127.0.0.1:5000/image/gradcam |
| Example Response | **Success**:  {"success": true,  "size": [224, 224, 3],  "image": "/9j/4AAQSk...8v1rqkhJan/2Q=="} |
| Response Notes | size → size of the image (for this endpoint its always simply 224,224,3  image → base64 encoded image: It’s the input image with a gradientcam image overlayed) |